REMARKS

Claims 1-8, 11-25, 28-42 and 45-56 are pending in the present application. By this Response, claims 1, 2, 8, 14, 18, 19, 25, 31, 35, 36, 42 and 48 are amended, claims 9, 10, 26, 27, 43 and 44 are canceled, and claims 52-56 are added. Claims 1, 18 and 35 are amended to recite monitoring execution of the computer program to identify events associated with one or more graphical user interface components, generating during execution of the computer program, contextual information for the identified events, wherein the contextual information provides information defining a context in which text associated with the one or more GUI components is presented, and providing an editor for editing the text in the displayed visual representation. Claims 8, 25 and 42 are amended to include subject matter similar to canceled claims 9, 10, 26, 27, 43 and 44, respectively. Claims 8, 25 and 42 are further amended to recite replaying generation of the user interface feature based on the description as associated with the item of text and providing an editor for editing the item of text during replaying of the generation of the user interface. Claims 14, 31 and 48 are amended to recite receiving edit instructions, via an introspective editor, during reproducing of the system events, to thereby edit text used in a user interface presented by execution of the executable of the computer program during reproduction of the system event. Claims 2, 19 and 36 are amended to be consistent with the amendments to claims 1, 18 and 35, respectively.

Support for the amendments to claims 1, 2, 8, 14, 18, 19, 25, 31, 35, 36, 42 and 48 may be found at least at page 6, line 8 to page 7, line 1 and page 12, line 12 to page 13, line 25. Reconsideration of the claims in view of the above amendments and the following remarks is respectfully requested.

I. Telephone Interview

Applicants thank Examiner Blackwell for the courtesies extended to Applicants' representative during the September 28, 2004 telephone interview. During the telephone interview, Applicants' representative discussed the above amendments to the claims and the distinctions of the amended claims over the cited references. Examiner Blackwell stated that Applicants' representative provided him with a clearer understanding of the invention and that the amendments better define the invention. However, Examiner Blackwell stated that he needed to review the references more closely before making a final determination as to whether the amended claims define over the cited references. The substance of the telephone interview is summarized in the following remarks.

II. <u>35 U.S.C. § 102, Alleged Anticipation</u>

The Office Action rejects claims 1-6, 14, 18-23, 31, 35-40 and 48 under 35 U.S.C. § 102(b) as being allegedly anticipated by Hinks et al. (U.S. Patent No. 5,678,029). This rejection is respectfully traversed.

A prior art reference anticipates the claimed invention under 35 U.S.C. § 102 only if every element of a claimed invention is identically shown in that single reference, arranged as they are in the claims. *In re Bond*, 910 F.2d 831, 832, 15 U.S.P.Q.2d 1566, 1567 (Fed. Cir. 1990). All limitations of the claimed invention must be considered when determining patentability. *In re Lowry*, 32 F.3d 1579, 1582, 32 U.S.P.Q.2d 1031, 1034 (Fed. Cir. 1994). Anticipation focuses on whether a claim reads on the product or process a prior art reference discloses, not on what the reference broadly teaches. *Kalman v. Kimberly-Clark Corp.*, 713 F.2d 760, 218 U.S.P.Q. 781 (Fed. Cir. 1983). Applicants respectfully submit that Hinks does not identically show every element of the claimed invention arranged as they are in the claims, as discussed hereafter.

Hinks is directed to a system and method for translating software into localized versions. With the system of Hinks, a data-centric approach to translating text of a program from one language to another is used in which a translation table is utilized to perform the translation. Source resource files are parsed by an export/import module to identify translatable strings which are stored in the translation table. The translation table encapsulates all the information that is know or can be derived from the various resources and stores them in a format which may be utilized by editors, e.g., string editor, menu editor, dialog editor, and the like. The editors are used to translate the strings to thereby generate translations which are themselves stored back in the translation table. The translated text is then merged back to the sources by generating a translated resource file.

The target product may then be rebuild with the translated resource file being a new source (column 7, line 8 to column 8, line 37).

Thus, Hinks operates on static source resource files. Hinks parses these static resource files to identify text to be translated, populates a translation table with this text and then permits a user to edit this text in the translation table using an editor. Hinks does not monitor the execution of a computer program to identify events associated with one or more graphical user interface components. Moreover, Hinks does not generate, during execution of a computer program, contextual information for the identified events, wherein the contextual information provides information defining a context in which text associated with the one or more GUI components is presented. Nowhere in Hinks is there any teaching or suggestion that Hinks operates on any executing computer program. This is because Hinks is directed to static analysis and translation of resource files whereas the claimed invention, as recited in claim 1, is directed to monitoring the dynamic execution of a computer program in order to obtain contextual information which may then be used to edit text associated with one or more graphical user interface components.

While Hinks may provide information regarding the GUI elements, this information is not obtained from monitoring an execution of a computer program as in the presently claimed invention, as recited in claim 1. To the contrary, Hinks operates only on the source code resource files and is not capable of generating contextual information during execution of a computer program for events identified during execution of the computer program.

In fact, Hinks describes the editors that are used starting at column 12. As described, the string editor, which is used for bulk translation of strings, provides information that includes information about the location where the message appears, its maximum length, its context information, and other such information which assists the user with translating the message (column 12, lines 61-66). The "context information" mentioned here includes an Identifier, a type, an original string, a new string, a comment, and filename (column 12, lines 36-41). This is information that is obtained statically from the resource files. There is nothing in the context information mentioned in Hinks that teaches or even implies that the context information is obtained from monitoring an executing program to identify events and then generating the contextual information for the identified events while the program is executing.

Furthermore, Hinks states that contrary to the string editor, which provides bulk translation of strings, the menu editor provides "live" translation of menus, i.e. the look and feel of the menu is created on screen while the end user translator is carrying out the translation (column 13, lines 19-26). Hinks further states that the dialog editor provides a similar "live" image of both the source and target dialog boxes (column 14, lines 27-29). While Hinks states that a "live" translation of the menu may be generated during editing of the menu, this still does not teach or even suggest that Hinks monitors the execution of a computer program to identify events or generates contextual information, while the computer program is executing, for the identified events. Furthermore, from the alternative nature of the language in column 13, lines 19-26, it is clear that the string editor of Hinks is not capable of providing a "live" translation and is provided solely as an offline bulk translation mechanism.

Thus, Hinks does not teach, or even suggest, the features of claim 1. In addition, claims 18 and 35 recite similar features and are thus, allowable over Hinks for similar reasons as set forth above. Therefore, Applicants respectfully submit that the rejection of claims 1, 18 and 35, and their respective dependent claims, should be withdrawn.

Regarding independent claims 14, 31 and 48, these claims recite receiving edit instructions, via an introspective editor, during reproducing of the system events, to thereby edit text used in a user interface presented by execution of the executable of the computer program during reproduction of the system event. Hinks does not teach an introspective editor, let alone receiving edit instructions via an introspective editor during reproducing of systems events that occurred during an execution session of a computer program as recorded in a sequential record.

The term "introspective editor" is specifically defined in the present specification at page 13, lines 7-9 as follows "An introspective editor is a library, that when compiled into an executable program, <u>allows a user to edit GUI features while the program is executing</u>" (emphasis added). Hinks does not teach or suggest any such editor. As discussed above, the editors in Hinks, i.e. the string editor, menu editor, and dialog editor, operate offline, i.e. when the computer program is not executing. This is clear in that

Hinks states that the editors store the translated text/menus/dialogs into the translation table which is then used to generate translated resource files so that the "product" may be rebuilt with the translated resource files. Hinks does not operate on an executing program and thus, does not provide an introspective editor that allows a user to edit GUI features while the program is executing.

Moreover, Hinks does not teach or even suggest receiving edit instructions via an introspective editor, during reproducing of system events that occurred during an execution session of a computer program, as recorded in a sequential record. Again, Hinks only teaches parsing resource files to populate a translation table, providing editors for the entries in the translation table, and then storing translations back into the translation table in order to generate new resource files. Hinks has nothing to do with reproducing system events that occurred during an execution session of a computer program, let alone providing an editor for editing GUI features during reproduction of such system events. Thus, Hinks does not teach or suggest the features of independent claims 14, 31 and 48.

In view of the above, Applicants respectfully submit that Hinks does not teach each and every feature of independent claims 1, 14, 18, 31, 35 and 48 as is required under 35 U.S.C. § 102(b). At least by virtue of their dependency on claims 1, 18 and 35, respectively, Hinks does not teach each and every feature of dependent claims 2-6, 19-23 and 36-40. Accordingly, Applicants respectfully request withdrawal of the rejection of claims 1-6, 14, 18-23, 31, 35-40 and 48 under 35 U.S.C. § 102(b).

III. <u>35 U.S.C. § 103, Alleged Obviousness</u>

The Office Action rejects claims 7-13, 15-17, 24-29, 30, 32-34, 41-47 and 49-51 under 35 U.S.C. § 103(a) as being allegedly unpatentable over Hinks in view of Liu et al. "Enhancing a GUI Event Recorder to Support the Creation of User Documentation.". This rejection is moot with regard to canceled claims 9, 10, 26, 27, 43 and 44 and is respectfully traversed with regard to the remaining claims.

Regarding dependent claims 7, 15-17, 24, 32-34, 41, 49-51, these claims are dependent upon respective ones of claims 1, 14, 18, 31, 35 and 48 and thus, are allowable over Hinks for at least the same reasons as set forth above with regard to these independent claims. Moreover, Liu does not provide for the deficiencies of Hinks.

Liu is directed to a mechanism for using a "Listen" tool to record low-level events in a GUI in order to construct a single sequence task model for generating user documentation. Specifically, the mechanism of Liu is used to generate on-line help documentation for a computer program. The "Listen" tool, when activated, records, in a list, the actor, verb and actee of GUI actions as illustrated on page 5 of Liu.

While Liu teaches a "Listen" tool for recording low-level events of a GUI, Liu has nothing to do with editing text used in a user interface of a computer program. Furthermore, Liu provides no teaching or suggestion that the "Listen" tool may be used to obtain context information that is then combined with text information to generate a visual representation of a GUI component such that an editor may be provided for edit the text in the displayed visual representation of the GUI component. To the contrary, Liu is directed to solving a completely different problem from that of either Hinks or the present invention. Thus, Liu does not teach or suggest the features of independent claims 1, 14, 18, 31, 35 and 48, and furthermore, is non-analogous art to the invention recited in these claims or even the invention recited in the primary reference, Hinks.

The present invention is directed to translating text in a user interface of a computer program. Hinks is also directed to translating text in a user interface of a computer program. However, Liu is directed to a system for generating user documentation based on task models generated from actor/verb/actee information for a GUI recorded by a "Listen" tool. Liu has nothing to do with translating text in user interfaces and does not teach or suggest that the mechanisms of Liu may be used for such a purpose.

Because Liu is non-analogous to Hinks, one of ordinary skill in the art would not have even attempted to modify Hinks to include features from Liu when presented with the problem of translating text in a user interface. To the contrary, there is no problem in Hinks for which the features of Liu are a solution or vice versa. Thus, there is no teaching or suggestion in Hinks to include any features from Liu. Moreover, there is no problem in Liu for which the mechanisms of Hinks would be a solution. This is because Hinks and Liu are directed to solving two completely different problems in two completely different ways. Other than mentioning graphical user interfaces, the Hinks and Liu references have nothing else in common.

Furthermore, even if it were somehow obvious to attempt a combination of Hinks and Liu, the result still would not be the invention as recited in independent claims 1, 14, 18, 31, 35 and 48. To the contrary, any alleged combination of Hinks and Liu would result in a system substantially as taught by Hinks in which user documentation for the GUIs is generated based on task models generated using the actor/verb/actee information recorded by a "Listen" tool. The resulting system still would not provide a method, apparatus, or computer program product in which an executing computer program is monitored to identify events for which contextual information is generated while the computer program is executing, as recited in claim 1, and similar features found in the other independent claims. Furthermore, the resulting system would not combine contextual information with text information to form a visual representation of a GUI component and then provide an editor for editing the text in such a visual representation, as recited in claim 1, and similar features found in the other independent claims.

Thus, Applicants respectfully submit that neither Hinks nor Liu, either alone or in combination, teach or suggest the features of independent claims 1, 14, 18, 31, 35 and 48 from which claims 7, 15-17, 24, 32-34, 41, 49-51 depend. Therefore, at least by virtue of their dependency, claims 7, 15-17, 24, 32-34, 41, 49-51 define over the alleged combination of Hinks and Liu. Accordingly, Applicants respectfully request withdrawal of the rejection of claims 7, 15-17, 24, 32-34, 41, 49-51 under 35 U.S.C. § 103(a).

With regard to independent claims 8, 25 and 42, these claims recite replaying generation of the user interface feature based on the description as associated with the item of text and providing an editor for editing the item of text during replaying of the generation of the user interface. Hinks does not teach these features. In fact, there is no teaching in Hinks of replaying anything, let alone the generation of a user interface feature based on a recorded description, recorded during execution of the computer program, in association with an item of text. Moreover, nowhere in Hinks is there any mention of editing an item of text during replaying of the generation of a user interface. To the contrary, Hinks only teaches statically parsing resource files, placing the text in a translation table, providing an editor for editing the text, storing the edited text in the translation table, and then generating translated resource files which may be used to rebuild the "target product" with the new resource files.

While Hinks teaches that certain menus and dialogs may be created during editing, the creation of these menus and dialogs is not based on a description of a user interface feature, whose generation is detected when executing the computer program, which is recorded during execution of the computer program. To the contrary, the menus and dialogs are created based on the information stored in the translation table which is information gathered from a static analysis of resource files. Thus, Hinks does not teach or suggest each and every feature of independent claims 8, 25 and 42.

In addition, Liu does not provide for the deficiencies of Hinks. The teachings of Liu have been discussed above. For similar reasons as set forth above, the alleged combination of Hinks and Liu, even if such a combination were possible and one of ordinary skill in the art were somehow motivated to make the alleged combination, would not result in the invention as recited in claims 8, 25 and 42. For example, Liu does not teach replaying generation of a user interface feature based on a description as associated with an item of text and then providing an editor for editing the item of text during replaying of the generation of the user interface. Since neither Liu nor Hinks teach or suggest these features, any alleged combination of Hinks and Liu still would not result in these features being taught or suggested.

In view of the above, Applicants respectfully submit that neither Hinks nor Liu, either alone or in combination, teach or suggest the features of independent claims 8, 25 and 42, from which claims 11-13, 28-29 and 45-47 depend. At least by virtue of their dependency, the alleged combination of Hinks and Liu does not teach or suggest the features of dependent claims 11-13, 28-29 and 45-47. Accordingly, Applicants respectfully request withdrawal of the rejection of claims 8, 11-13, 25, 28-29, 42 and 45-47 under 35 U.S.C. § 103(a).

IV. <u>Newly Added Claims</u>

Claims 52-56 are added to recite additional features of the present invention. Specifically, claim 52 is added to recite a context interpreter that generates a visual representation as a recreation of the text in the context described by the contextual information without using the computer program to generate the visual representation. Support for the features of this claim may be found at least at page 6, lines 8-18 of the present specification. Claims 53 and 55 are added to recite a scripting shell that replays an execution session of a computer program and an introspective editor being provided during replaying of the execution session of the computer program. Support for this claim may be found at least at page 6, line 19 to page 7, line 1 of the present specification. Claims 54 and 56 are added to recite that the introspective editor is provided in response to user input halting replaying of the execution session. Support for this claim may be found at least at page 17, lines 20-24 of the present specification. Neither Hinks nor Liu, whether taken alone or in combination, teach or suggest the features of newly added claims 52-56.

V. Conclusion

It is respectfully urged that the subject application is patentable over Hinks and Liu and is now in condition for allowance. The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

Respectfully submitted,

DATE: October 1, 2004

Stephen J Walder

Stephen J. Walder, Jr.
Reg. No. 41,534
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 367-2001
Attorney for Applicants